A METHODOLOGY OF
DESIGN FOR MANUFACTURABILITY
OF
FIBER LAMINATE COMPONENTS

BY

ANAND SINGH

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1992

A METHODOLOGY OF DESIGN FOR MANUFACTURABILITY OF FIBER LAMINATE COMPONENTS

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by

ANAND SINGH

to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
APRIL, 1992

ii

13.4.92

CERTIFICATE

This is to certify that present work entitled, "A Methodology of Design for Manufacturability of Fibre Laminate Components" has been carried out under our supervision and that it has not been submitted elsewhere for the award of a degree.

Dr. J.L.Batra

Professor,
Mechanical Engineering
and
Industrial & Management Engineering,
I.I.T.Kanpur.

Dr. S.G.Dhande

Professor,
Mechanical Engineering
and
Computer Science & Engineering

I.I.T.Kanpur.

April 6, 1992.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

I acknowledge with sincerity and deep sense of gratitude the expert guidance, continuous encouragement and valuable suggestions provided by Dr.S.G.Dhande and Dr. J.L.Batra during the course of this work.

I am thankful to all my friends and colleagues for their help at various stages of the work. Particularly I would mention of Tarun, Mukul, Sreenivas, Piyush, B.S.S.Pradhan and Amul.

The present work is sponsored by ADA Bangalore. I take this opportunity to acknowledge ADA for their support. I would like to thank Mr. Prakash and Mr. Shimha for their useful suggestions.

I am indebted to CAD Project laboratory and its staff for the facilities provided to me and their kind cooperation.

- Anand Singh

LIST OF FIGURES

The following figures pertain to the Illustrative Example in Chapter 4.

LIST OF SYMBOLS

A	-	Automatic Nesting
CP	-	Cut Piece
$\underset{\sim}{g}$	-	Generatrix of a Cylinder
I	-	Interactive Placement
k	-	Gaussian Curvature
k_{min}, k_{max}	-	Principal Curvatures
M	-	Number of Pieces in Buffer
N	-	Total Number of Cut Pieces in a Particular Tape Width
$\underset{\sim}{n}$	-	Normal to a Surface
n(i)	-	Number of Pieces of Width w(i)
P	-	A Set Containing 4 Instances of a Piece
p(u)	-	Curve in X-Z Plane
r_u	-	Tangent Vector in u Direction
r_v	-	Tangent Vector in v Direction
s^a, s^b	-	Length of Projectors
u, v	-	Parameters of a Surface
UD	-	Unidirectional Tape
w	-	Width of a Ply in a Layer
W	-	Total Width of Stock
w(i)	-	Width of ith Piece

ABSTRACT

The strength of a composite component not only depends on the material properties and thickness, but also on the fibre orientations of each layer in the component. Therefore, composite components are designed as a stack of layers with preferred orientations. Generally, composites (with the fibers of materials like graphite, boron etc.) are available in the form of pre-impregnated tapes of constant widths. Therefore the process of manufacturing involves laying up of series of strips adjacent to one another to form various layers. Several such layers are then formed one over the other.

In manufacturing large number of components, it is an immense task to keep record of each layer and its strips. This necessitates the use of computer based methodologies for manufacturing.

Present work is an effort to computerize the process of design for manufacturability with a view to save the costly composite material during the manufacturing of components. Surface of the component is unwarped into flat patterns. Then strips (cut pieces) data is generated for various layers which is then nested on the tapes so as to minimize the wastage of material when pieces are cut from the tapes. Finally, the geometrical data for transfer foils, which are used for transferring the cut pieces to the component mould, is generated. This methodology has been implemented in the form of an interactive software package, CANES-UD. Illustrative example showing a specific case that can be handled with software is described in the present work.

Chapter 1

INTRODUCTION

1.1 Introduction to Composites

Composites are defined as material consisting of two or mor constituents having chemically distinct interfaces. The majo constituents are fiber and matrix. The fibers are embedded in th matrix and are principal reinforcing and load carrying agents i composites. They are strong and stiff. The matrix can be organic metallic or ceramic. The function of the matrix is to support an protect fibers and provide a means to distribute load betwee fibers.

Unlike most of engineering materials composites are non homogeneous and non-isotropic. Objects made of composite generally consist of several distinct layers. A layer may diffe from another in the following ways :

- relative volumes of constituents,
- form of reinforcement and
- orientation of fibers.

Due to all above mentioned properties, there is a uniqu flexibility in the design of composite objects. It becomes ver easy to manipulate required strengths in various directions. I the following paragraphs some of the methods for manufacturin composite components have been discussed.

Hand Lay-up

It is the simplest and oldest of the composites fabricatio processes. It is especially used for large components like boa

hull. Reinforcing mat is positioned manually in open mould and resin is poured, brushed or sprayed over. Entrapped air is removed manually with squeezes or rollers (Figure 1.1).

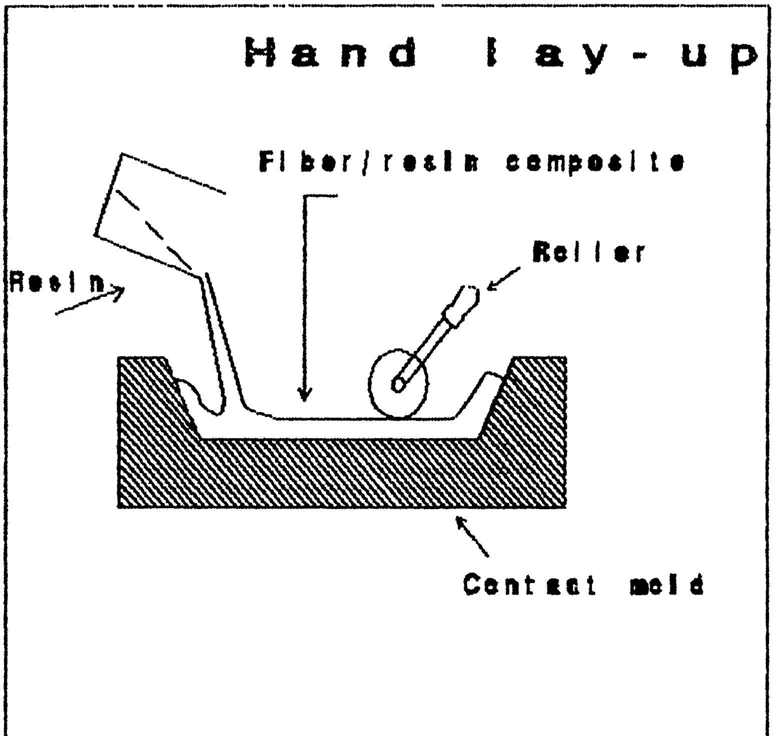

Figure 1.1

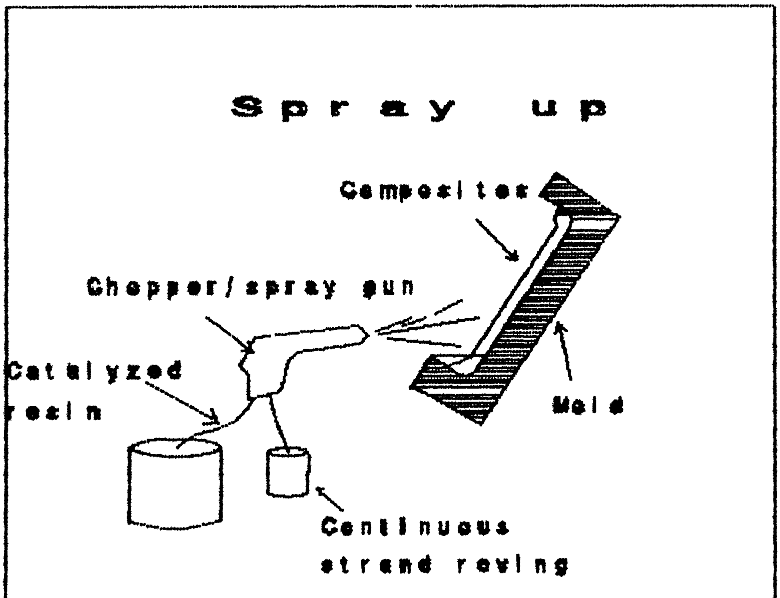

Figure 1.2

Spray Up

It is a partially automated form of hand lay up. Chopped glass fibers and resins are simultaneously deposited on an open mould. Fiber glass roving is fed through a chopper on spray gun and blown into a resin stream which is directed at mould. Woven roving is often added in specific areas of greater strength (Figure 1.2).

Vacuum Bag Lay-up

The vacuum bag molding process is a refinement of hand lay up. In this process, vacuum is used to eliminate entrapped air and excess resin. After lay-up is made, a non adhering film (usually nylon) is placed over lay-up and sealed at the edges. Vacuum is drawn onto the bag formed by the film and composite is cured at room temperature. This provides higher reinforcement concentration and better adhesion between layers [1] (Figure 1.3).

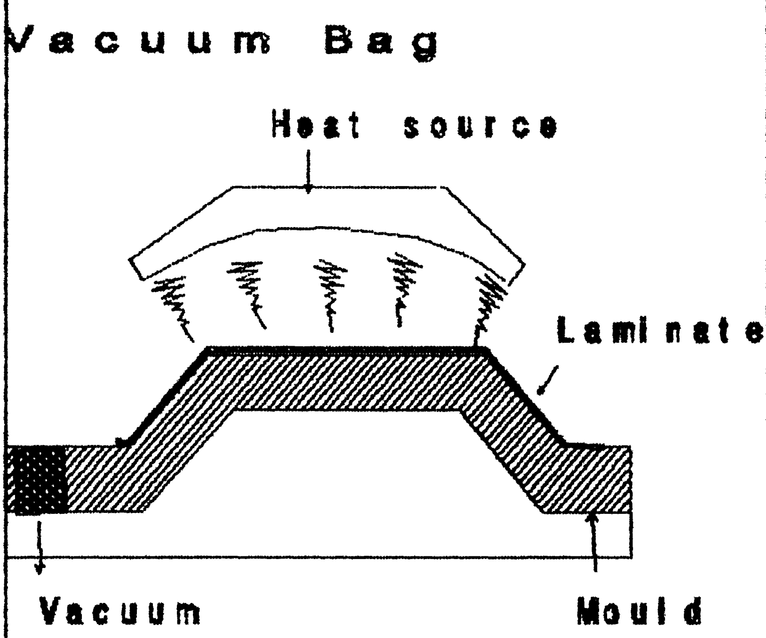

Figure 1.3

Autoclave

Autoclave molding is a modification of pressure bag and vacuum bag molding. This method produces denser, void free molding because higher heat and pressures are used in cure. Autoclaves are essentially heated pressure vessels into which bagged lay-up is taken for cure cycle (Figure 1.4).

roduction

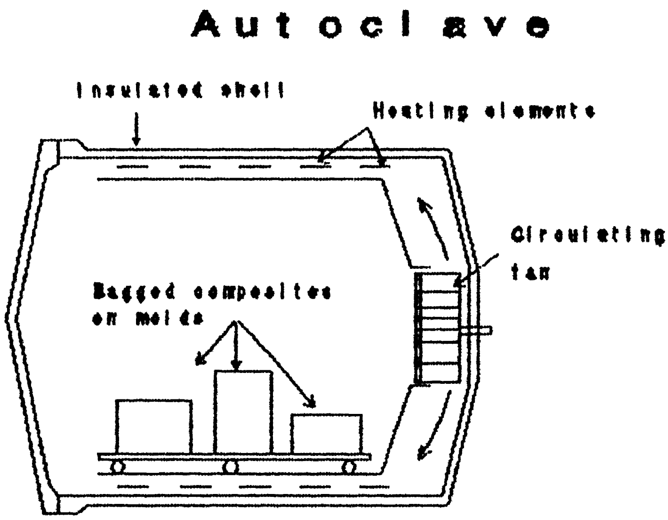

Figure 1.4

Filament Winding

High speed precise lay down of continuous reinforcement in pre-described patterns is the basis of filament winding method. This technique is used for manufacture of surfaces of revolution such as pipes, tubes, cylinders and spheres etc.

Pre-Formed Molding Compounds

A large number of reinforced thermosetting products are made by preformed molding compounds or premix to which all necessary ingredients have been added. This enables faster production rates to be achieved. Prepreg is another term for preformed molding compounds. Prepregs consist of roving, woven fabric, continuous unidirectional fiber reinforcement sheets or random chopped fiber sheets, impregnated with a partially cured resin system. Most of the prepreg use epoxy resins. The prepregs after curing are sandwiched between two layers of release film, such as silicone

impregnated paper or polyethylene film. All this is done prior to winding it on a roll.

Before molding prepreg is cut to required size and both layers of polyethylene films are removed. The prepreg cut pieces are then placed into the mould. Subsequently, they are pressed and cured [2].

1.2 State of Art

Composite materials are now establishing themselves as a useful engineering material. From fuel efficient cars to stealth aircraft, the modern hi-tech way to build structures is by using composites. Design of composite components, like an aircraft wing, consists of several layers of composite material stacked in different orientations. The fiber orientations in each layer are dictated by design considerations. A layer is made up of one or more plies or zones, which have well-formed polygonal geometries. The process of manufacture of such components involves cutting of number of pieces from, unidirectional (UD) composite material tapes, of standard widths.

When manufacturing hundreds of components in a batch (which is generally the case in aircraft wing design), accounting for each layer and each ply becomes a highly cumbersome and involved task. Such tasks necessitate the use of computer aided methodologies. The accounting for layers or layer management includes an elaborate coding scheme. This scheme involves coding of plies to designate their material and fiber orientation, sequence in ply lay up and, time of release from refrigerating storage (as composite material is perishable in untreated state) [4].

In computer aided manufacturing approach (Figure 1.5), first the surfaces are configured and then unwarped into flat configurations (flat patterns). At this point ply (zone) geometries and ply numbers are specified and data for ply lay up is recorded. The ply data is then given to a nesting program which generates geometric data for cutting plies optimally out of available composite tapes. This data is fed to a NC X-Y table with a rapidly reciprocating razor knife operating vertically, and a vacuum system below to hold the raw material in place. Ply strips cut out by the process are placed into mould using transfer foils. After suitable heat treatment on the mould final desired component is obtained [3].

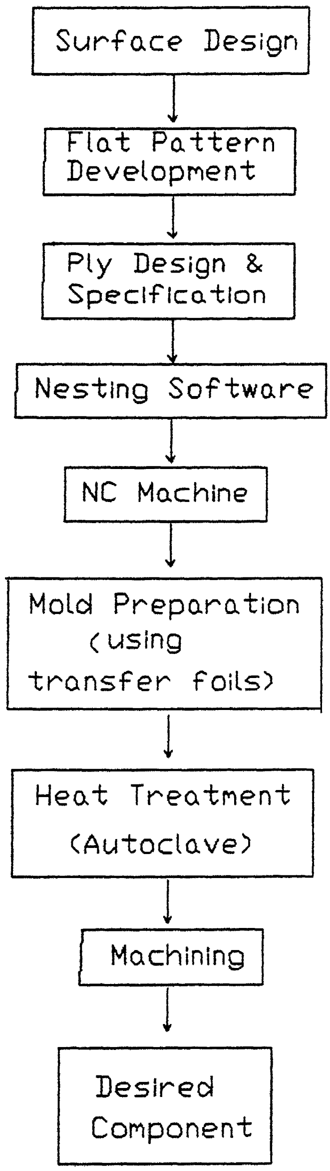

Figure 1.5 : Process chart for manufacturing a composite component

1.3 Computer Aided Manufacturing Approach

Composites are often available in the form of pre-impregnated fibers "Prepregs". They are held in position not only by epoxy matrix but by a removable backing that prevents the tape from sticking together in roll. These prepregs are available in rolls of standard widths.

Since the size of resource (i.e. tapes) is generally less than the size of layers (i.e. flat patterns), process of manufacturing of components involve cutting of large number of pieces from unidirectional (UD) tapes of standard widths and then use these cut pieces to obtain final component (Figure 1.6).

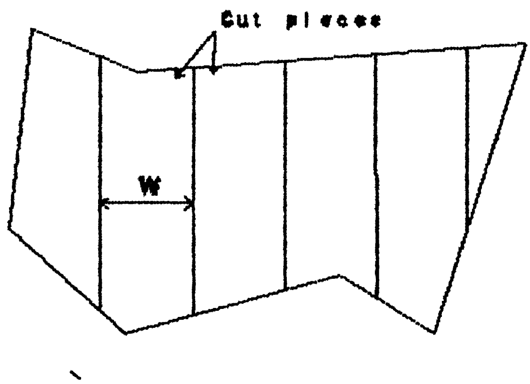

Figure 1.6

If this process is done manually there is a lot of wastage of costly composite material and also it becomes difficult to handle such huge data. Therefore it is highly desirable to use a computer based methodology to enhance productivity of manufacturing. In the following few paragraphs, an approach has been discussed for computer aided manufacturing of composite components.

As mentioned earlier that there are a number of layers in a component and each layer is again made up of several consecutive

strips which form several lines within the layer. While laying subsequent layers care should be taken to avoid strips with boundaries matching boundary lines of previous layer. Otherwise stacking up of boundaries along same line weakens the component. This constraint is similar to the one used in laying of bricks in the construction of a wall.

Pieces are cut out from UD tapes, to generate various layers. To minimize wastage in this process, pieces are first nested along the tapes of corresponding widths. Now these nested pieces can be cut from the tapes using a knife or a NC cutter.

Placement of these cut pieces back into required form i.e. in mould, is a difficult task. A slight shift causes formation of air pockets in the component. To avoid this many techniques have been designed and the most popular one is the Transfer Foil technique [15]. In this process, mylar sheets are taken. These mylar sheets are transparent, polyester flexible sheets. Prepreg ply strips are laid on one face of transfer foil assembly. The foils have reinforced holes that are coordinated with pegs in the mould. This assumes proper orientation and location of mylar sheets. Various steps required to make transfer foil assembly are as follows :

• Selection of axis and orientation of the foils so as to have smallest possible transfer foil assembly, required for a layer,
• Drawing ply strips on the transfer foil assembly,
• Drawing location of rivets and peg holes on the foils and
• Drawing individual transfer foils with the portion of ply strips contained in them.

1.4 Objectives of Present Work

Many components of an engineering system, particularly in an aircraft, are made from composite materials. In the present work, an approach has been discussed for computer aided manufacturing of composite components. Main objective here is to automate the process of design for manufacturability with a view to save the costly composite materials during manufacturing of parts.

For the complete process of computer aided manufacturing of composite components various processes like surface design, flat pattern development, strip cutting, unidirectional nesting and generating data for transfer foils need to be automated. Our objective is to develop a user-friendly software implementing all the above mentioned processes. This software can also be useful in many similar applications where resource size is smaller than the object size (flat patterns).

1.5 Literature Review

Literature in the form of technical reports etc. is available for carrying out the work (i.e. manufacturing of composite components) manually or semi-automatically. However, literature related to the complete automation is very scarce or deliberately kept latent by industries for competitive reasons.

Sufficient literature is available on the sub processes like nesting and transfer foil method. This has been reviewed briefly as follows.

Considerable research has been done in the area of nesting of irregular shapes on rectangular domains. The general optimal

nesting problem remains unsolved, however, optimal solutions for simple geometries are well known. Unfortunately it is the general problem which has practical significance since in most of the cases the shapes to be nested have non-regular geometry.

Researchers have adopted many techniques for 2-dimensional nesting of irregular shapes on a rectangular domain.

Albano and Adamwicz [11] used heuristic techniques. Irregular shapes are first allocated in rectangular enclosures. Second step is the optimal nesting of these enclosures using various heuristics. The solution can be improved using an interactive system.

Segencich [12] used a simple approach of sliding a piece to a suitable position till there is no overlap with already allocated pieces. Pieces to be nested are selected interactively. Overlapping is checked by mapping the shape with a matrix of numbers; two integers are assigned, one for boundary of the shape and other for interior. At every step, matrix is checked if there is an overlap with other shapes. Summation of matrix elements at that point is indicative of an overlap.

Chow [13] developed an algorithm for nesting shapes on a strip (in one column). The pieces should have one half of the boundary as duplicate of the other half. These identical portions of boundary make it possible for neighbouring shapes to mate together without gaps.

Amarel [14] uses an algorithm of sliding vectors for interactive placement of pieces. A sliding vector is calculated which gives the direction and magnitude in which pieces being placed should be moved to avoid collision.

Shama Rao [15] has explained the significance and methodology

of transfer foil process in transferring of cut pieces to the component mould.

Chapter 2

CAD OF FIBRE LAMINATE COMPONENTS

2.1 Basic Problems

This chapter discusses in brief the basic issues or problems in the manufacturing of composites components and these are discussed in detail in subsequent chapters.

Conventional manufacturing processes like casting, machining, welding etc. are totally uncommon in the manufacturing of composite components because of their different chemical composition. Various design steps for composite components are:

• Definition of the geometry.

• Defining the thickness along the surface. This depends upon the structural properties of the component.

• Defining the component as an assemblage of a set of layers of composite fibers. Each layer will have a finite thickness. Some layers may have holes or disjoint polygons. For each layer a specific orientation direction has to be specified. All this design is based on structural considerations as well as information about fibre properties of composite materials.

This completes the component design but does not give any information about design for manufacturability. In case of composites the later part involves several steps.

Carbon fibre composites are available in the form of unidirectional tapes of constant width. If the demand i.e. layer size is greater than the resource i.e. composite tapes then the process of manufacturing involves cutting of number of pieces from

JD tapes. The pieces placed together constitutes the original .ayer.

Nesting techniques are used to minimize the wastage while :utting pieces from UD tapes. Finally all the pieces are :ransferred layer by layer to a mould. Curing of the cut piece ıssembly in autoclave makes the component in its final shape.

2.2 Geometric Design of Surfaces

The first step in the design of a composite component is the definition of its surface (Figure 1.5). This surface can be planar surface or a curved surface. Since the composite material is available in the form of flat sheets/tapes, it requires that surface to be manufactured be unfolded and laid onto a plane first. This process is called development of surfaces.
The present section describes different types of three dimensional surfaces which are used to define the geometry of components. Section 2.3 explains various surface development methods in detail.

Surfaces can be broadly classified as :

- Ruled surfaces,
- Double curved surfaces and
- Free form or sculptured surfaces.

Ruled Surfaces

When a straight line, called generatrix, moves while in contact with one or more straight or curved lines, called directrices, so as to form a surface then a ruled surface is obtained. In some ruled surfaces, the generatrix moves such that

it is not only in contact with a directrix/directrices but also remain parallel to a plane called director [6].

Ruled surfaces can further be classified as

(i) Plane surfaces.
(ii) Single curved surfaces.
(iii) Warped surfaces.

(i) Planar Surfaces

These are generated by moving a straight line generatrix in a plane. Their parametric representation is

$$\mathbf{n} \cdot \mathbf{r} = 0 \qquad ..(2.1)$$

where $\mathbf{n}$ is the normal to the curve.

(ii) Single Curved Surface

Single curved surfaces are generated by moving a straight line generatrix such that any two consecutive portion of it are either parallel or intersecting. Examples of single curved surfaces are cylinders, cones, convolutes etc.

Parametric representation of a cylinder is

$$\mathbf{r}(u,v) = \mathbf{r}_1(u) + v\,\mathbf{g} \qquad ..(2.2)$$

Cones are represented parametrically by

$$\mathbf{r}(u, v) = (1 - v)\, \mathbf{r}_1(u) + v * \mathbf{r}^{v} \qquad ..(2.3)$$

$$0 \leq v \leq h$$

$\mathbf{r}^{(v)}$ is the position vector of vertex of cones and $\mathbf{r}_1(u)$ is the directrix curve describing the base of the cone.

(iii) Warped Surfaces

Warped surfaces are generated by moving a straight line generatrix such that any two consecutive positions of it are non-co-planar. However, these consecutive positions are parallel to the director plane.

Double Curved Surfaces

If a constant or variable generating curve is moved along another curve then a double curved surface is obtained. Double curved surfaces are classified as (i) surfaces of revolution (ii) surfaces of general form.

(i) Surfaces of Revolution: These are obtained by revolving a plane curve around an axis lying in its plane. The curve is called profile curve and in its various positions around axis, it creates meridians. For the simplest case, let z axis be the axis of rotation and let the curve

$$p(u) = x(u) + z(u)$$

be defined in X-Z plane. Then equation of the surface of revolution is

$$p(u, \theta) = x(u)\cos(\theta) + x(u)\sin(\theta) + z(u) \qquad ..(2.4)$$

Examples of surfaces of revolution are sphere, prolate, ellipsoid, paraboloid, hyperboloid, annular torus etc.

(ii) Surfaces of General Form: These are obtained when the generating curve moves along a general curved path [6].

Free Form or Sculptured Surfaces

Although analytical surfaces, e.g., ruled surfaces, discussed in previous section, are important for design and manufacturing, they are not sufficiently flexible for many applications. Free form surfaces provide higher flexibility by using higher degree of polynomials for both patch boundary curves and the interior blending functions. Free form surfaces are generated by combining the patches of this nature. Various example of free form surfaces are Ferguson´s cubic surface patch, Coon´s patch and Bezier´s unisurf surface [6].

2.3 Development of Surfaces

Development is the process of laying out or unfolding a surface into a plane. A developed surface may consists of solely a single developable or a set of piece-wise developable components. Developable surfaces are of considerable interest for industries where objects are fabricated out of a sheet.

Double curved and free form surfaces are non-developable. In ruled surfaces only single curved surfaces are developable.

Condition of Developability

To determine if a surface or a portion of surface is developable it is necessary to consider the curvature of parametric surface.

At any point P on the surface, the curve of intersection of a plane containing the normal to the surface at P and surface, has a curvature ´k´.

As plane is rotated about normal, curvature changes. Euler

showed that there are two unique directions for which curvature is maximum and minimum. The curvatures in these two directions are called principal curvatures k_{min}, & k_{max}
Now gaussian curvature is defined as

$$k = k_{min} * k_{max} \quad ..(2.5)$$

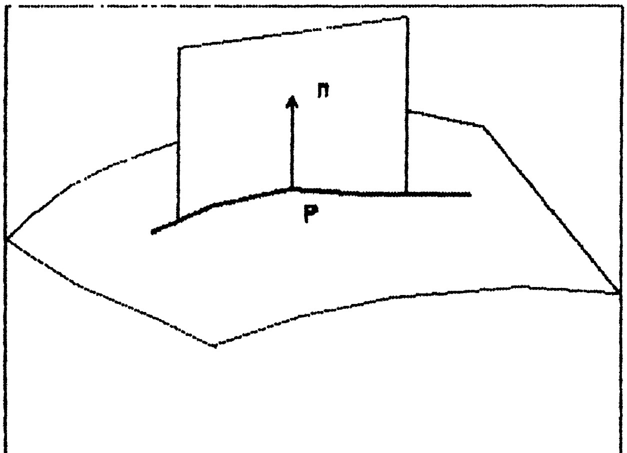

Figure 2.1

For a biparametric surface gaussian curvature is also defined as

$$K = \frac{A.C - B^2}{| r_u . r_v |^4}$$

where ..(2.6)

$$A = [r_u . r_v].r_{uu}$$
$$B = [r_u . r_v].r_{uv}$$
$$C = [r_u . r_v].r_{vv}$$

$$r_u = \frac{dr(u, v)}{du}$$

$$r_v = \frac{dr(u, v)}{dv}$$

Condition for developability of a surface is that the gaussian curvature is zero everywhere on the surface [5].

Methods of Development

Depending upon surface to be developed different methods development are used. Parallel line method, radial line method and triangulation are essentially the three methods used [7]. Parallel line method is used to develop surfaces of cylinders, prisms etc. where all rulings are parallel to each other. If rulings form a set of intersecting lines as in case of cones and pyramids, then radial line method is used. Triangulation method is used for piece-wise development of surfaces. Any single curved ruled surface can be developed by this method.

However, these manual methods are time consuming and are subject to drafting inaccuracies. To reduce time consumed and inaccuracies, development process has be automated. For this various suitable mathematical models have been developed and are available in literature on analytical and differential geometries [8, 9, 10].

A ruled surface Σ_1 can be mapped onto a planar surface Σ_2, which is called development of Σ_1

This mapping can be isometric, isogonal or isoareal. In isometric mapping length of infinitesimal arc on one surface and length of its image on other are equal. Similarly in isogonal mapping angles are preserved and in isoareal mapping, the area is preserved. It should be noted that the development of surface is the process of isometric mapping [8].

2.4 Summary Observations for Manufacturing

A composite component is geometrically considered to be a surface with some finite thickness. The thickness may be constant

over the entire area or may not be constant. Generally depending on the loading conditions and support conditions the component thickness is designed so as to ensure satisfactory performance in static and /or in dynamic conditions. A structural analysis is usually performed for ensuring the safety of the component.

The strength of a composite component not only depends on the material properties and the thickness, but is also dependent on the orientation of each layer in the component. As such one can think of a composite component to be a stack of several layers. Each layer will have a pre-designed direction of orientation. In order to provide desired strength, it is necessary to have different orientations for adjacent layers of a component. In practice the structural engineer in consultation with composite material specialist designs this layout scheme of the different layers of a component. In short the design of the composite material component involves not only the geometrical design of the surface and the thickness of surface but also involves the design of sequencing and orienting of different layers of the component. In case of composite material component the process of manufacturing is neither of a material removal type (e.g. milling, turning etc.) nor it is of material transformation type (e.g. casting, forging etc.). The process of manufacturing involves laying up of series of strips adjacent to one another to form a layer. Several such layers are formed one on the top of other. Such process of manufacturing requires certain considerations of design for manufacturability to be taken into account. These considerations involve the problem of strip layout, transfer foils technique and nesting of cut pieces on given strip of constant width. The next chapter describes how the methodologies of automation are developed for these problems.

Chapter 3

CAM OF FIBRE LAMINATE COMPONENTS

3.1 Strip Design

Pre-impregnated composite materials are available in the tapes of standard widths. The process of generating a ply consists of cutting of a number of cut pieces (or ply strips) from available tapes. These ply strip when placed together will constitute the original ply. In order to cover a ply, selection of composite tapes, out of various available tapes, is done in such a way that minimum number of ply strips are required. This is done to keep the wastage of material minimum and also to make handling and accounting of ply strips easier.

The strip design problem is basically a simple knapsack problem [17]. A knapsack problem can be explained using a simple example. Say there are n different types of articles available. Each article of type 'i' has a weight 'wi' kg and has a value 'vi' rupees. A box that can hold a weight of at the most 'w' kg has to be loaded with these articles so as to maximize the total value of the articles included. There can be number of ways of placing weights into the box. But for this problem it is not possible to develop a non-heuristic algorithm to solve it. These type of problems are known as NP complete problems [17]. Though it is not possible to find a global optimal solution for such cases, it is possible to get a good feasible solution by applying some constraints (bound and branch approach) and then to compute a near optimum objective value.

The strip layout problem discussed here is also of knapsack ype. There is a stock of width W and w(i) width pieces are enerated out of it. n(i) is the number of pieces of width w(i).

$$\sum_{i=1}^{\text{width types}} n(i)*w(i) \leq W$$

$$n(i) \geq 0 \qquad \text{and integer for all } i$$

ue to material handling considerations viewpoint minimum number of ieces should be generated. In addition to this, there are taggering lines constraints (as mentioned in Section 3.3). Then ollowing equations are solved to get a solution.

$$\sum_{i=1}^{\text{width types}} n(i)*w(i) \leq W$$

$$n(i) \geq 0 \qquad \text{and integer for all } i$$

$$\sum_{i=1}^{\text{width types}} n(i) \qquad \text{should be minimum}$$

he main steps of strip cutting algorithm are as follows:

tep1 : Read the geometry data i.e. the geometry of plies or zones resent in the layer (Figure 3.1).

tep2 : Orient the layer geometry such that fiber orientation in he layer is along the Y-axis. This is done to make allocation long the width on horizontal axis.

Step3 : Round-off the width to an integral multiple of Basic Unit Size (BUS). Basic Unit Size is the minimum of the various available tape widths. This step is valid if all the tape widths are integral multiple of smallest tape width.

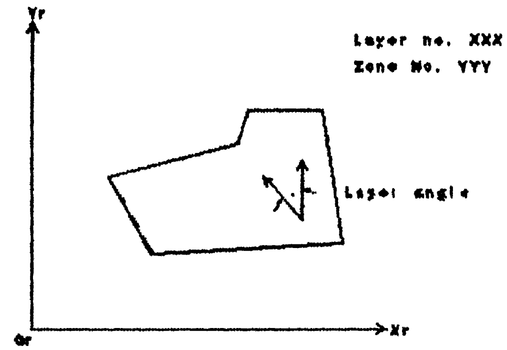

Figure 3.1

Step4 : Impose a grid of vertical lines of BUS on the plies in the layer. Also impose the lines to be staggered. Lines to be staggered are nothing but edges of the strips of previous layer. For this, only those layers which have same fiber orientation as of current layer are considered.

Step5: Allocate Strips

Algorithm for allocation of tapes of widths x1, x2, x3 etc. for a width 'w' is as follows.

```
allocate for w
if (w < Ø) then allocation over;
while (all tapes are not tested)
      x = next largest tape from (x1, x2, ...)
      if( stagger lines not violated )
            save allocation x
            allocate for (w-x)
      endif
endwhile
```

Step6 : Forming Cut-Pieces

Cut the various zones in the layer at the boundary lines of allocated tapes. Use Cohen-Sutherland clipping algorithm for dividing the polygon across various lines.

Cut pieces generated in this process of strip layout will have a width equal to any of the available tape widths. These pieces are termed as 'standard pieces'. But there may be certain pieces which do not have widths equal to any of the tape widths. These cut pieces are termed as 'non-standard' or 'miscellaneous' pieces.

3.2 Cut Piece Nesting

3.2.1 Introduction to nesting

Nesting is an allocation of two dimensional profiles on available stock, with a constraint to avoid overlaps between them, so as to minimize the total area of stock utilized for allocation. The nesting problem, in its purely geometrical form, finds application in many industries like sheet metal, garment, leather, composites. For example, in blanking parts from strips, cutting a leather piece for making handbags, jackets, shoes etc., cutting cloth pieces for garments, cutting of metal sheets for sheet metal applications. Other applications include cutting of rectangular sheets of steel, wood, glass etc., cutting of paper board for production of boxes. Ship building and railway coach making involve cutting of irregular shapes from large sheets of metal.

This optimization (nesting) problem has been solved for a long time manually which needed more time if number of the shapes requiring nesting are large. Computer nesting methods are preferred over manual methods because they are much faster and more

accurate. In mass production, material cost is usually a large portion of total cost because of large production volumes; even a small percentage saving in material will result in a significant money saving.

Method of nesting must be compatible with the material properties and tool characteristics. Some materials have different properties on each of the two faces. For such materials, the shape can be rotated in the plane of material, however out of plane rotation (mirror images) are not allowed. Some materials such as cloth, wood, composites etc. have grain direction thereby limiting rotation angle to Ø & 18Ø degree.

Although lot of work has been done on 2-D nesting of patterns, there is very little literature available for 1-D nesting. Chow has published a paper [13] on nesting along a strip but he assumes a particular symmetry in the geometry of the patterns to be nested. Albano [11] has used dynamic programming and heuristic approach but it will not be economical to use these for 1-D nesting, if computer time is considered to be important.

In the present work, a sliding vector approach as mentioned in [13] has been used to develop a simple algorithm for 1-D nesting of cut pieces. Using this algorithm, it is possible to compromise between nesting efficiency and computer time. Following section discusses the algorithm in detail.

3.2.2 UD - Nesting

After generating the geometrical data of cut pieces (see Section 3.1), the next step is to arrange these pieces on UD-tapes so that when the material is cut out of tapes, the wastage is minimum.

A pattern or cut piece is described as a simply connected polygon. Patterns considered in the present work are non self intersecting polygons; whose boundary is composed of only straight line segments. These polygons may have polygonal holes in them. Nesting of available patterns on a UD-tape is 1-D nesting (a special case of 2-D nesting). Here the width of patterns is equal to width of the stock. This limits the rotation of patterns to Ø and 18Ø degrees only, however they can be flipped. Thus in all there are four ways of placing a pattern on the UD-tape. In other words each pattern evolves an enumerated set of 4 instances.

$$P = \{ P_i^1, P_i^2, P_i^3, P_i^4 \} \qquad ..(3.1)$$

Therefore in order to allocate N patterns, 4 * N instances have to be analyzed to find best pattern. But as N increases the marginal gain in nesting efficiency cannot compensate for increase in computer time. Therefore a buffer of M patterns (M <= N) is

$$\begin{aligned} Buffer = \{ P_i^j \} \\ i = 1 \ to \ M \\ j = 1, 2, 3, 4 \end{aligned} \qquad ..(3.2)$$

created

Out of a buffer of 4 * M instances, the best instance is selected for placement. Then enumerated set corresponding to selected pattern P, is flushed from buffer and next set corresponding to an unallocated pattern is taken into the buffer. The above process is repeated till all patterns are allocated.

3.2.3 Criterion Selection of Best Pattern

Best pattern is that which gives minimum wastage area with a

already allocated pattern. Waste area is the portion of stock/tape left unused between two allocated patterns. Purpose of nesting is to minimize this area (hatched portion in Figure 3.2). Selection of B with respect to A depends on the magnitude of this area.

3.2.4 Steps to Calculate Waste Area

Here the waste area between two patterns is found when they are placed butting at their edges.

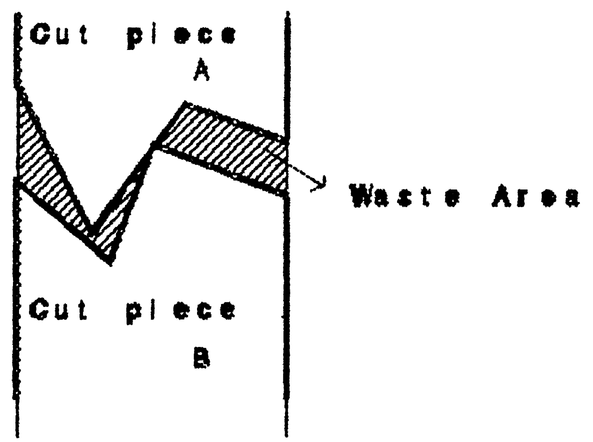

Figure 3.2

(i) Place the patterns with their bounding rectangles butting each other.

(ii) However, this matching of rectangles does not ensure closest packing. Therefore B is made to slide towards A until they strike each other with any overlap [13].

(iii) Calculate the Strike Distance i.e. the distance by which B slides towards pattern A.

The **Strike Distance** is the minimum of the strike distance 1 and strike distance 2. (Figure 3.3), where

strike distance 1, s^a, is the length of projector from lowest point of pattern A on pattern B; and the

strike distance 2, s^b is the length of projector from highest point of pattern B on pattern A

(iv) Slide B by strike distance.

(v) Compute the magnitude of the hatched area (Figure 3.2).

3.2.5 Interactive Placement

The nesting module should have the facility for interactive placement of patterns on the tape.

Two things should be displayed on the screen

(i) UD tape with automatically nested patterns, and

(ii) Unallocated patterns.

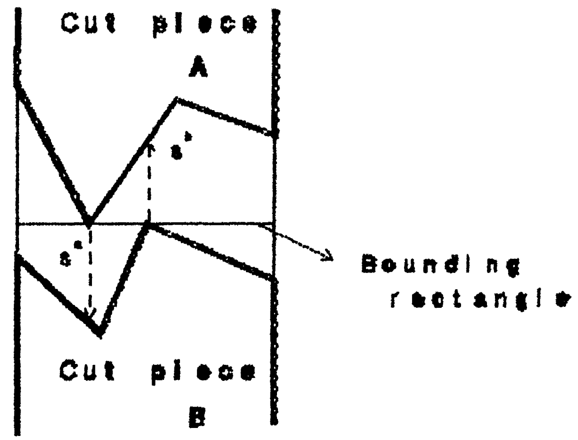

Figure 3.3

Interactive feature allows user to utilize any waste zones of the tape. These are the possible sites for unallocated patterns. Using a locating devise (mouse), user picks up the unallocated pattern and places in an unallocated zone in the tape. However program should help user in deciding patterns, which can be feasibly allocated inside the waste pockets.

3.2.6 Pseudocode for Nesting Algorithm

```
    input standard tape widths
                                                  read and store cut pieces
                                                  cut piece is abbreviated as CP
    while  (till all cut pieces are finished)
        input  vertices of cut piece
        change coordinates of CP from local to global
        width = CP maxX - CP minX                 cut piece width
        if (width == any of available tape widths) then
```

```
            store piece in store of calculated width
        else
            store CP in miscellaneous piece store
        endif
endwhile
```

Automatic nesting for a particular tape width

```
select  width for automatic nesting
select  buffer
n = buffer
m = number of cut pieces of selected width
border = tape edge
read first n unallocated CP's in a buffer
generate 4 instances of each CP in buffer
while (till all CP's of selected width are allocated)
        calculate waste area for all the instances, in buffer,
        with the working border
        find the instance which gives minimum waste area
        allocate the instance on tape
        store the above instance
        set ALLOCATION_DONE true for the CP corresponding to
        selected instance
        flush out all 4 instances, corresponding to selected CP,
        from buffer
        if (CP's not finished in store)
            read next unallocated CP from store
            store all 4 instances of selected CP into the buffer
        endif
```

```
endwhile
```

after automatic nesting on a tape some of the unallocated pieces can be accommodated interactively in the waste pocket on the tape.

Interactive Placement

```
select width of the pieces to be nested interactively

select one cut piece

 if (for selected piece ALLOCATION_DONE != TRUE)

      select mode of placement of CP

      if (there is no overlap with a already allocated CP's on
          tape)then

           place CP in a gap interactively using a locator
           device

           add CP in the database of allocated pieces on tape

      endif

endif
```

3.3 Transfer Foil Method

It has been explained in Section 2.4 that a composite structure is made up of a number of layers. A layer may have one or more than one plies (or zones) as shown in Figure 3.4. In a layer, the fibers are oriented in a particular direction.

The basic operations required to fabricate a composite component are :

- Production of individual plies.
- Placement of plies into mould.
- Preparation of lay-up and cure.

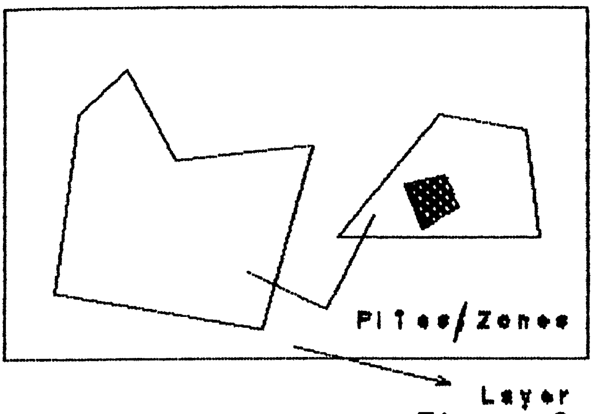

Figure 3.4

These operations look simple but their complexity would become obvious when actual requirements defined are considered.

The individual plies have different shape, size and fiber orientations. The fiber orientations are usually Ø, 9Ø, 45, -45 degrees. These are controlled with respect to a datum. Plies are made of number of ply strips which have to be oriented accurately, while placing them into the mould. Gaps and overlaps have to be controlled during butting of adjacent strips, otherwise air pockets are created in the component. Any attempt to adjust a strip by peeling off the strip will result in total rejection of already laid plies, due to tackiness of resin. Wrinkles have to be avoided which occur while correcting the alignment of ply strips [15].

To avoid all above mentioned problems, Transfer Foil method is most commonly used.

Transfer Foil Method of Lay-up :

In this method Mylar sheets are used for transferring ply strips to tool/mould. Therefore sheets are also called transfer foils.

Various desired properties of these sheets are as follows.

• Transfer foil sheets should be transparent and should have one face shiny and other dull. The dull face is used for plotting and shiny face is used for prepregs (strips) laying.

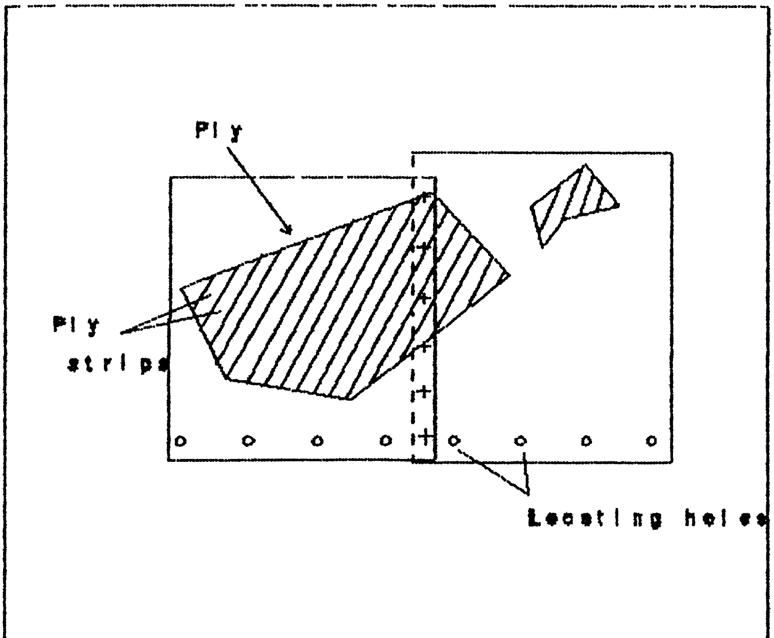

Figure 3.5

• Transfer foil sheets should be flexible enough to take contours of tool/mould.

• Transfer foil sheets should be moisture resistant and dimensionally stable.

The transferring operation is done layer by layer. Dimensions of transfer foils are found such that they cover all the plies of a layer (Figure 3.5). Depending upon the size of a layer, it may require one or a combination of several transfer foils. Holes are punched in the transfer foils at peg positions. When these holes are matched with pegs in mould, plies will be transferred to the required position in the mould [15].

If there are more than one peg axis in the tool, peg axis nearest to the plies is used for locating transfer foils assembly (Figure 3.6). Once a peg axis is selected, then an orientation is found out such that minimum number of foils are required to cover all plies in the layer (Figure 3.7). The purpose of selecting nearest peg axis and a proper orientation is to find the minimum possible size of transfer foil assembly. This makes handling of transfer foils assembly very easy.

CAM of FLC

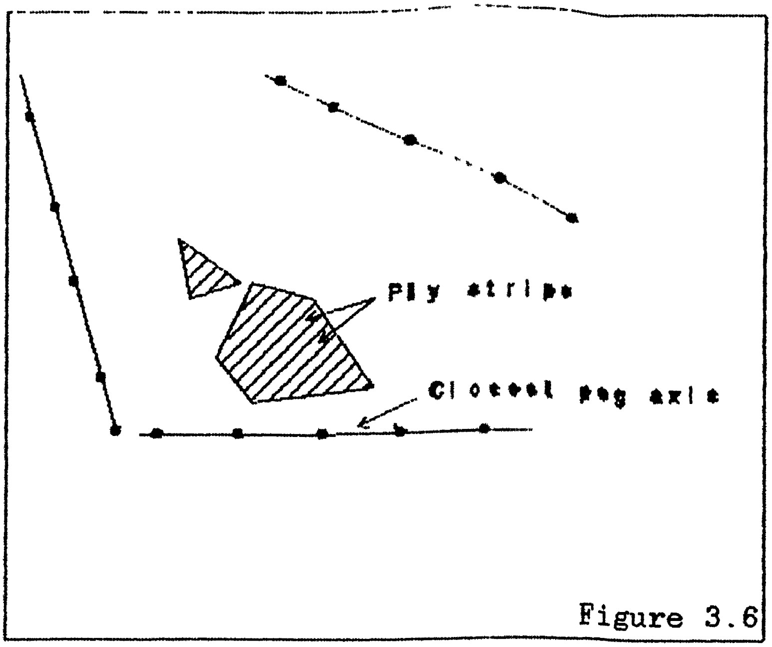

Figure 3.6

For placing or locating ply strips on the transfer foils assembly, they are first drawn on rough side of foils using a plotter. Next, the protecting coverings of strips are removed. Then strips are stuck on plane side of transfer foil assembly in positions and orientations as indicated by the drawings plotted on strip's rough surface. The whole assembly is then transferred, after flipping it, to the tool surface. In order to assure strong adhesive and airtight contact the strips assembly is pressed against the doubly curved tool surface with the help of special rollers. The rolling

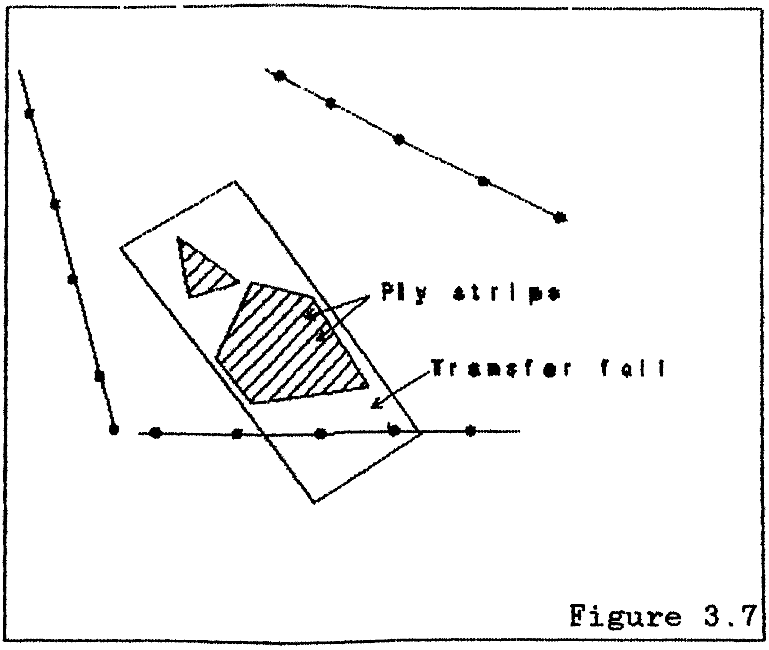

Figure 3.7

makes the binding between the tool surface and the strip surface stronger than the binding between the strips and the transfer foil. Finally to complete the operation the transfer foils, whose purpose has been served, are peeled off.

Pseudocode for Transfer foil method

```
Input     Available transfer foil widths
          Pegs positions in the mould
          Layer information

Select    peg axis nearest to the plies

Find      orientation w.r.t. selected peg axis for minimum
          width covered.

allocate transfer foils for the layer

allocation width 'w'

if (w <= Ø) then return  allocation over.

while (w > Ø)

     select a transfer foil (width = x) for which x >= w

     save allocation x

     w = w - x + OVERLAP
                                  (OVERLAP between two transfer foils)
endwhile

if (sufficient pegs are present in
     transfer foil assembly) then

     store transfer foil assembly coordinates
else

     calculate width of transfer foil assembly for
     sufficient pegs

     allocate transfer foil for the layer

endif

store rivet positions for each transfer foil
```

store portion of ply strips inside each transfer foil

store peg holes positions inside each transfer foil

3.4 Database Considerations

This section deals with the storage schemes used in the CANES-UD software for storing information about various layers, cut pieces, nested cut pieces, transfer foils etc. There is a hierarchy of data structures starting from a layer at the top to a cut-piece at the bottom. Layers are arranged as a linked list. Each layer consists of a list of zones, a list of UD-tape allocations for zones and a list of cut-pieces. Polygon list consists of fields, a pointer for a list of vertices for polygon number, a pointer to a list of holes and a pointer to cut-piece data. Both cut-pieces and zones are of the type 'polygon_list'. Zones, not being cut-pieces, have NULL value to cut-piece data field.

Vertex list is of type 'vertex_list'. It consists of a pointer to vertex of type 'vertex' and a pointer to next vertex. 'vertex' type consists of x, y, z fields for a point.

The information about various cut pieces generated during strip allocation is given to a nesting program, which nests the cut pieces along various tapes. The cut pieces data can be given to nesting program either through a file or it can be current working data.

For nesting program, the information about cut pieces are stored in an array of type 'data_type'. Information about the number of pieces and various utilization details corresponding to

each tape width is stored in an array of type 'width_store'

All above mentioned data types have been presented in Appendix I.

CHAPTER 4

CANES-UD SOFTWARE

4.1 Introductory Comments

CANES-UD is a graphic, menu driven package for subdividing polygons (with or without holes) across several unidirectional tapes of given widths. Once a polygon is distributed across strips, various smaller polygons are formed and all of them together constitute original polygon. All these small polygons are arranged on the basis of their widths; on standard available tapes. The aim of this arrangement (nesting) is to minimize the wastage of material when pieces are cut out from the tapes.

Another feature of the software is the generation of geometrical data for transfer foils, which are basically used for transferring various layers into the mould (Section 3.3).

Entire system is modular in structure and open-ended to the extent that additional features may be added in future without much difficulties and changes in the program. All modules have been written in 'C' and are implemented on SUN 3/60. SUNVIEW graphics package has been used for menus and panels etc. Therefore, basic graphic routines may need to be changed if the package is to be run on any other system.

Interactive Menu Features

The system provides interactive menus which help in going through the steps of input, design and display in systematic manner. When selection is to be made, user is given the list of

all the possible options. Also, default values have been set. The main advantage of this style of interaction is that user does not has to commit to memory any of the commands or values.

4.2 Input Specifications

The program requires a user to provide following information as input.

1. Information about component such as
   (i) Component Name,
   (ii) Component Number and
   (iii) Number of layers present in this component.

2. Information about each layer such as
   (i) Layer number,
   (ii) Number of zones in the layer,
   (iii) Geometry of zones (in term of lines) and
   (iv) Layer Angle.

   This is the orientation of entire layer measured from vertical axis. This is basically the direction of fibers in that layer. UD strips are laid along this direction only.

3. Layers for Stagger

   As mentioned above there are number of layers in a component and each layer is again made up of several consecutive strips which form several boundary lines within a layer. While laying subsequent layer care should be taken to avoid strips with boundaries matching with boundary lines of the previous layer. Otherwise stacking up of boundaries along same line greatly weakens

the component. Therefore, it is important to specify the number of layers for stagger. If this number is too high there will be too many boundaries which should be avoided for current strip layout. In that case, a feasible layout may not result at all.

4. Stagger Allowance

It is the distance from stagger boundaries (boundaries of previous layer) within which current strip layout boundaries should not fall.

5. Layers per Bundle

This is the number of layers which are processed in one go. If all the layers are processed at a time and cut pieces are nested, then cut piece inventory on the shop floor becomes so large that it is difficult to handle it. Therefore a certain number of layers are considered for one cycle.

6. Available Strip (Tape) Widths.

7. CANES-UD Geometry Files

CANES-UD has a DXF interpreter [AutoCAD reference manual, release 10]. DXF format is a data format in which AutoCAD generated drawings can be converted into readable ASCII text format. Therefore it is possible to make various drawings in AutoCAD and then transfer them to SUN workstation for their further processing, through CANES-UD software.

The program requires the user to provide following information as the input for transfer foil method.

CANES-UD Software

8. Available widths of Transfer foil.

9. Position of various pegs on the tool/mould. Pegs are used for locating transfer foil assembly on the tool.

10. Overlap between two transfer foils that is used for riveting.

11. Distance between two rivets. Rivets are used for joining two mylars.

4.3 Output Specifications

1. Cut piece (CP) database. Each record in this file should contain fields for values of Bundle number, Layer number, Zone number, Column number, CP width, a link list of vertices of CP.

2. Graphic display of CP's, Zones and Layers.

3. Component drawings of CP's with their code numbers.

4. Graphic display of nested CP's on UD tapes, also indicating CP's code number.

5. Resource Utilization

   % Utilization = (1 - Total Waste Area / Total Tapes area consumed)

6. Transfer (mylar) foil database for each layer. It contains number of foils required, a list of vertices of each foil, list of rivets and pegs present in each transfer foil.

7. Display of transfer foil assembly for a layer with various

zones present in the layer.

8. Display of individual foils for a layer, with the portions of various zones contained in them.

9. Output Geometry Files

The software has facility to convert all CP (cut piece) and transfer foil data into a DXF file of data. Thus users can get a hard copy output using AutoCAD.

4.4 Illustrative Example

The following section presents an example to lucidate the computer aided manufacturing approach for fibre laminate components, using CANES-UD software.

Input Data

(i) In this example, five layers are considered. Each layer has one or more plies (or zones). The geometrical data is given in Appendix II.

(ii) Layer angles for various layer have also been mentioned in Appendix II.

(iii) UD tape widths are as follows.

300 mm, 150 mm, 75 mm, 25 mm

(iv) Available transfer foil widths are 1200 and 1000 mm.

Results

(i) Figures 4.3, 4.5, 4.7, 4.9, 4.11 show strip layout for the plies present in the layers.

(ii) Figures 4.4, 4.6, 4.8, 4.10, 4.12 show the cut pieces

generated corresponding to each layer.

Total number of pieces corresponding to each of the available tape widths are as followed

Tape Width	Number of Cut Pieces
300 mm	68
150 mm	4
75 mm	6
25 mm	3

In addition to these there are 9 pieces of widths not equal to any of the available tape widths. These pieces are called non standard pieces.

(iii) Figure 4.13 shows automatically nested standard pieces with non standard pieces allocated interactively.

(iv) Figures 4.14, 4.16, 4.18, 4.20, 4.22 show transfer foil allocation for various layers.

(v) Figures 4.15, 4.17, 4.19, 4.21, 4.23 show the individual transfer foils for various layers.

Table 1 given below explains the various utilization details.

CANRS-DD Software

<table>
<tr><th></th><th>Tape Width (mm)</th><th>Tape length consumed (mm)</th><th>Total area of allocated CP´s (mm sq.)</th><th>% Utilization of Tape</th><th>Total % Utilization</th></tr>
<tr><td rowspan="4">**A**</td><td>300</td><td>71087.16</td><td>20958644.10</td><td>98.28</td><td rowspan="4">97.90</td></tr>
<tr><td>150</td><td>5165.54</td><td>674010.06</td><td>86.98</td></tr>
<tr><td>75</td><td>4309.07</td><td>320400.50</td><td>99.14</td></tr>
<tr><td>25</td><td>4015.04</td><td>99743.87</td><td>99.37</td></tr>
<tr><td rowspan="4">**A + I**</td><td>300</td><td>71928.148</td><td>21030154</td><td>**97.76**</td><td rowspan="4">94.64</td></tr>
<tr><td>150</td><td>5165.54</td><td>674010.06</td><td>86.98</td></tr>
<tr><td>75</td><td>4309.07</td><td>320400.50</td><td>99.14</td></tr>
<tr><td>25</td><td>7964.11</td><td>116894.09</td><td>**58.71**</td></tr>
</table>

A -> Automatic nesting. A + I -> Interactive nesting.

** Non standard pieces have been placed interactively on 300 and 25 mm tapes.

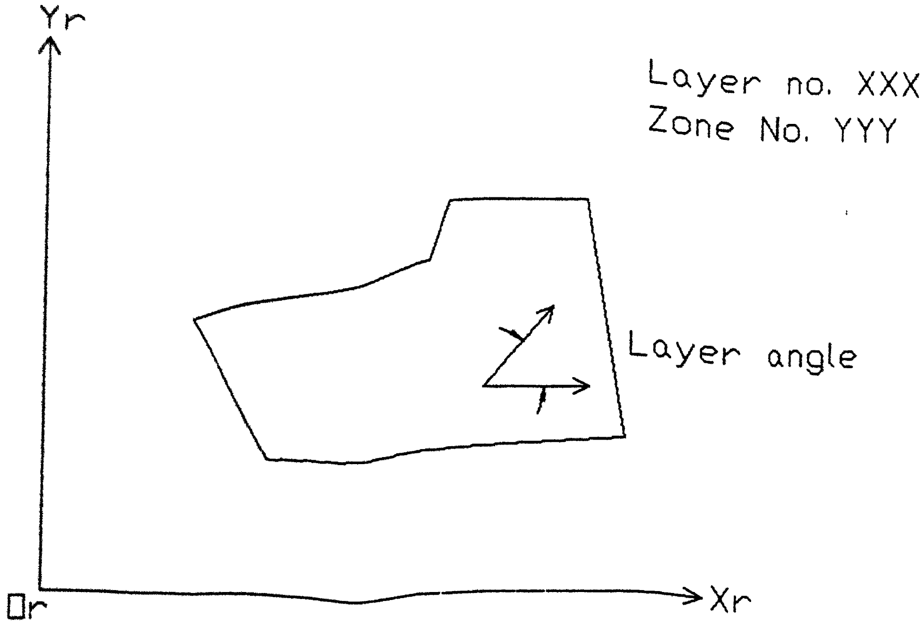

Figure 4.1 : Input Data

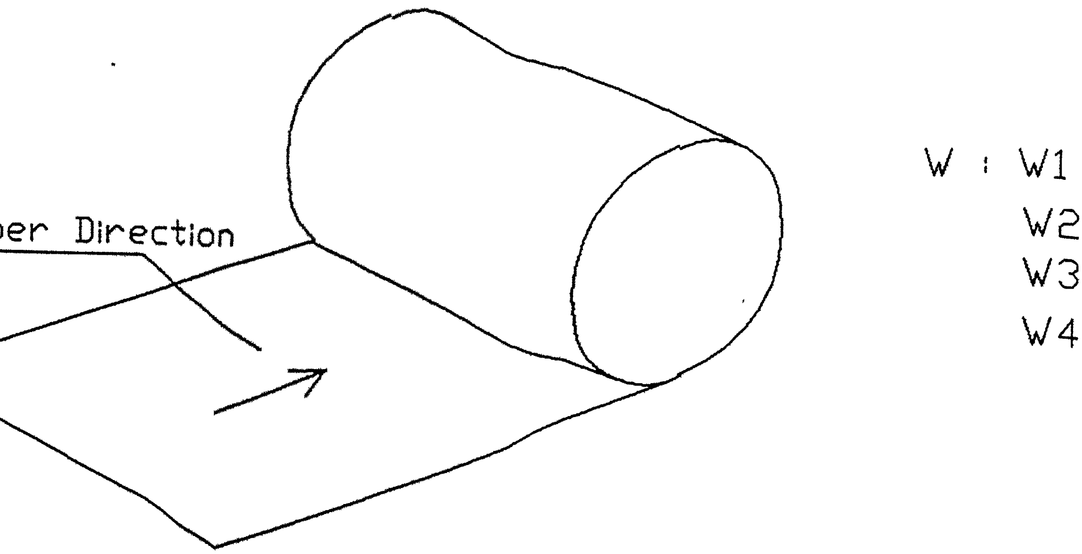

Figure 4.2 : Unidirectional Tape

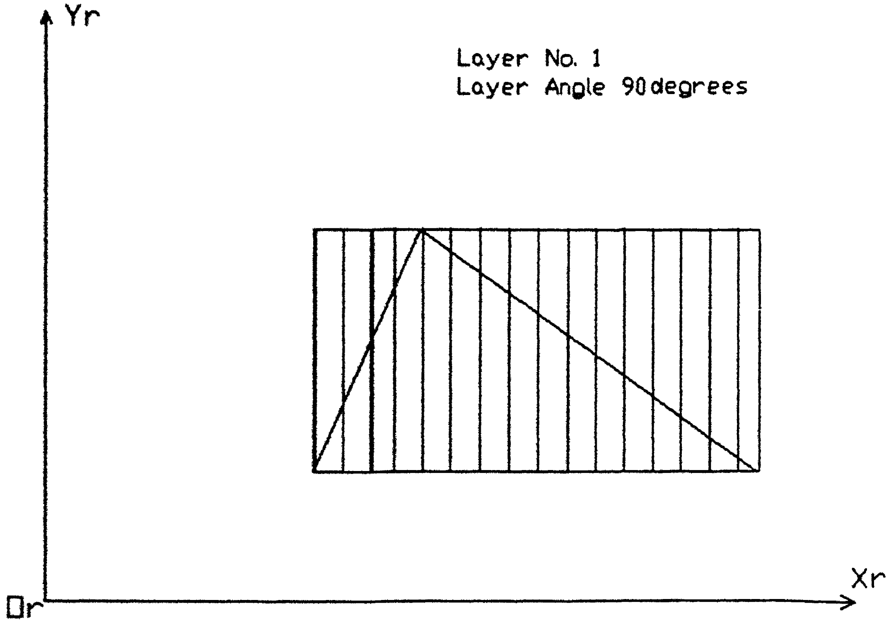

Figure 4.3 : Strip layout for layer no. 1

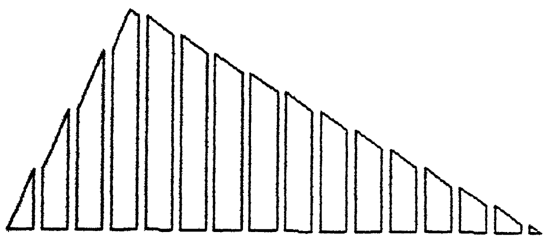

Figure 4.4 : Cut pieces for layer no. 1

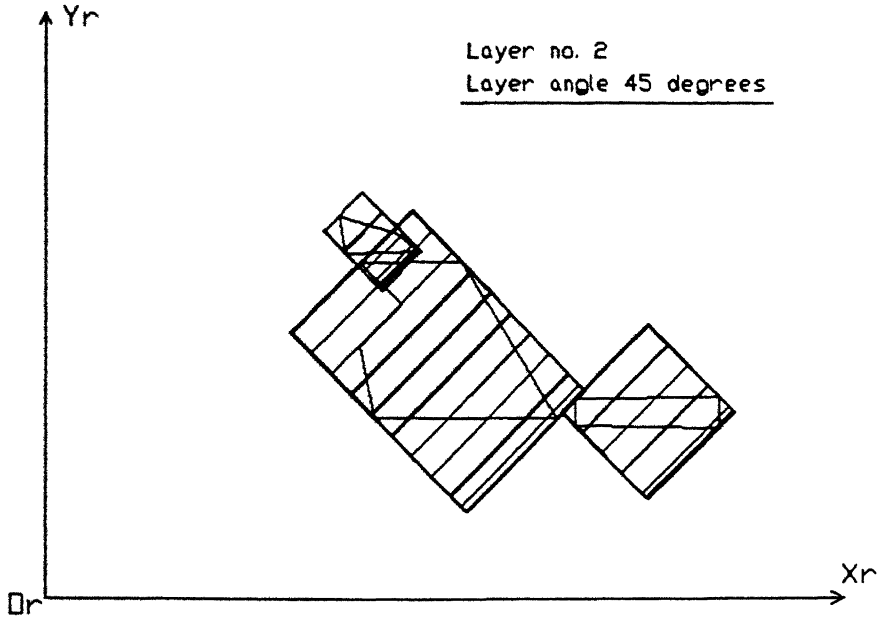

Figure 4.5 : Strip layout for layer no. 2

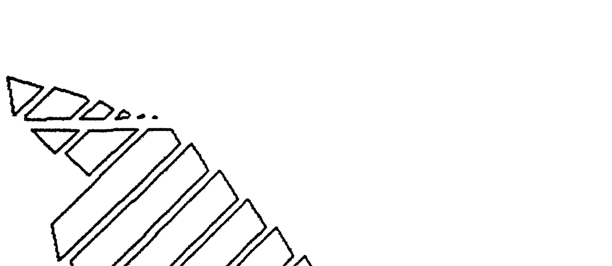

Figure 4.6 : Cut pieces for layer no. 2

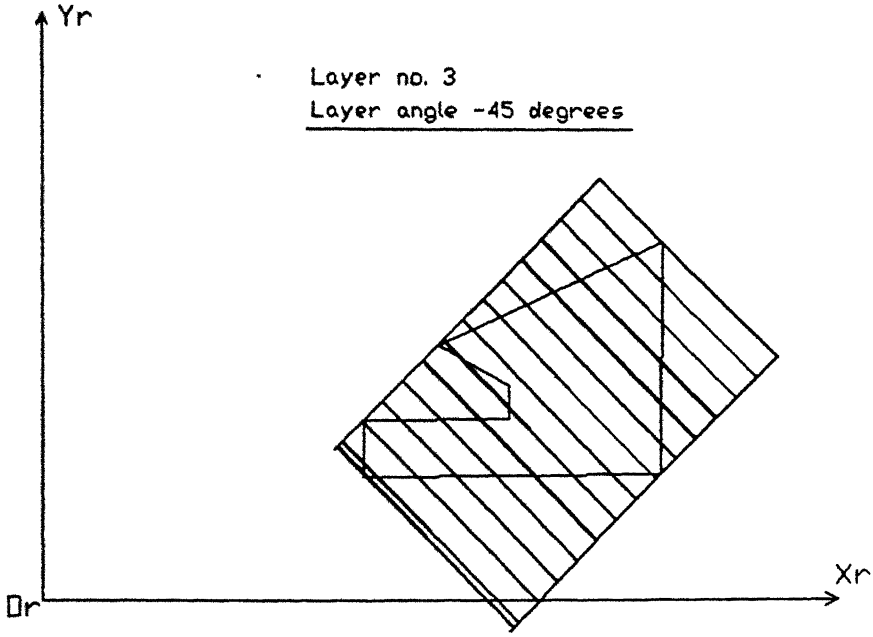

Figure 4.7 Strip layout for layer no.3

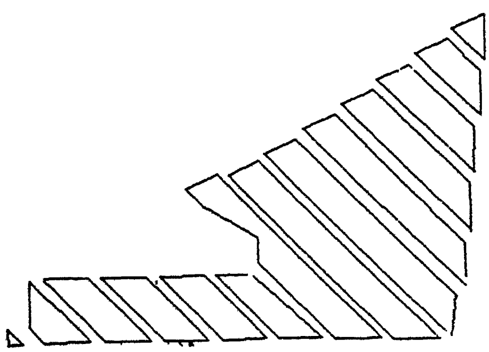

Figure 4.8 : Cut pieces for layer no. 3

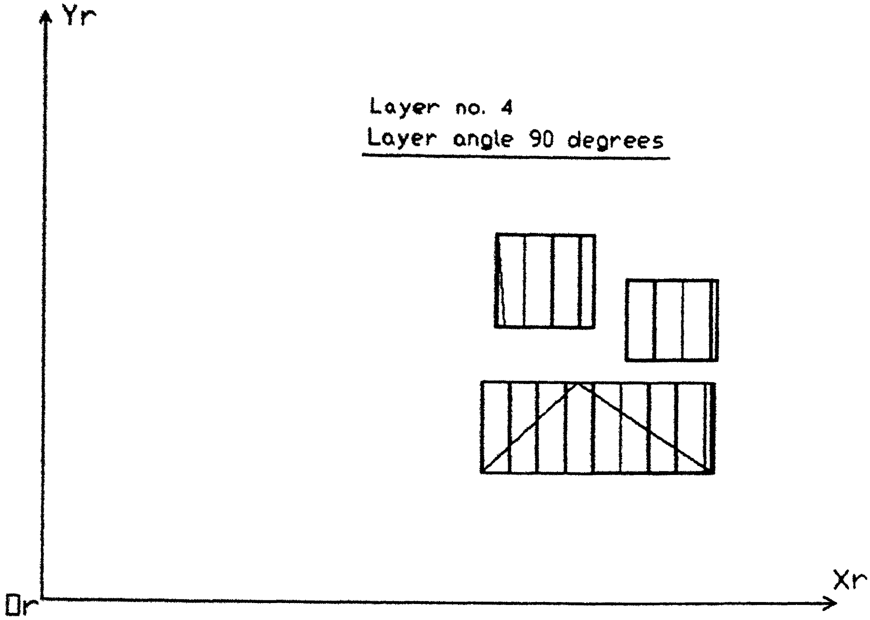

Figure 4.9 : Strip layout for layer no. 4

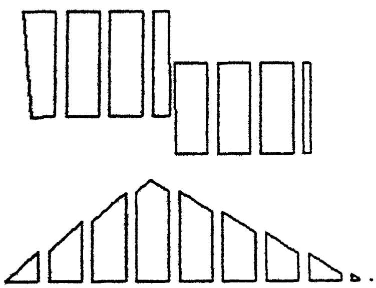

Figure 4.10 : Cut pieces for layer no. 4

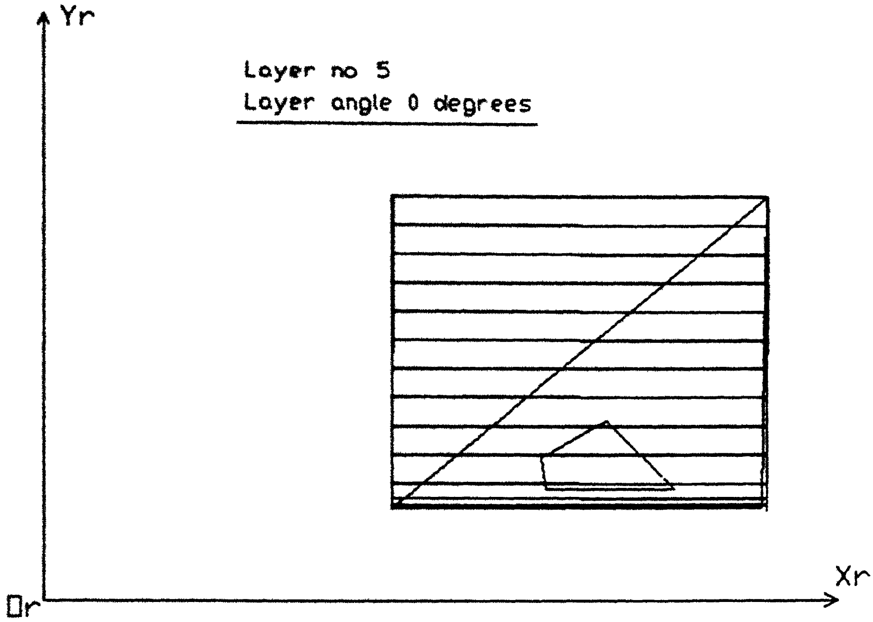

Figure 4.11 : Strip layout for layer no. 5

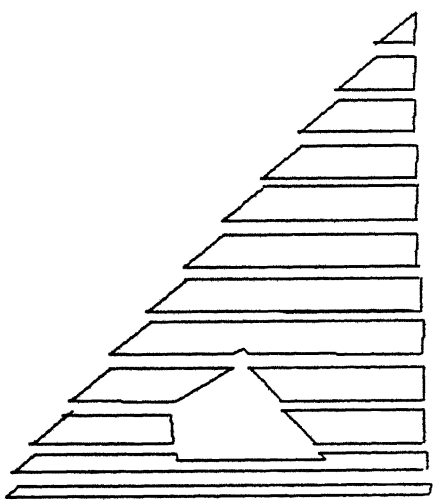

Figure 4.12 : Cut pieces for layer no. 5

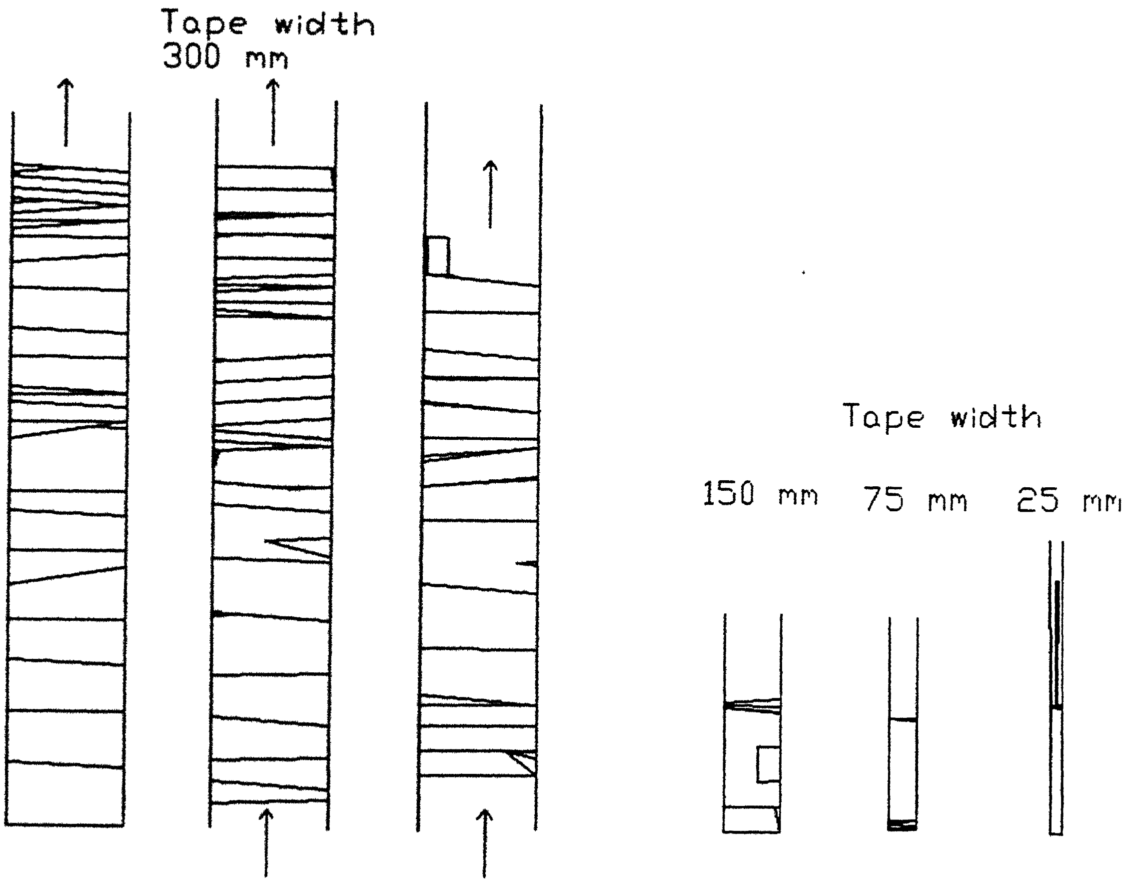

Figure 4.13 : Cut pieces nested along various composite t

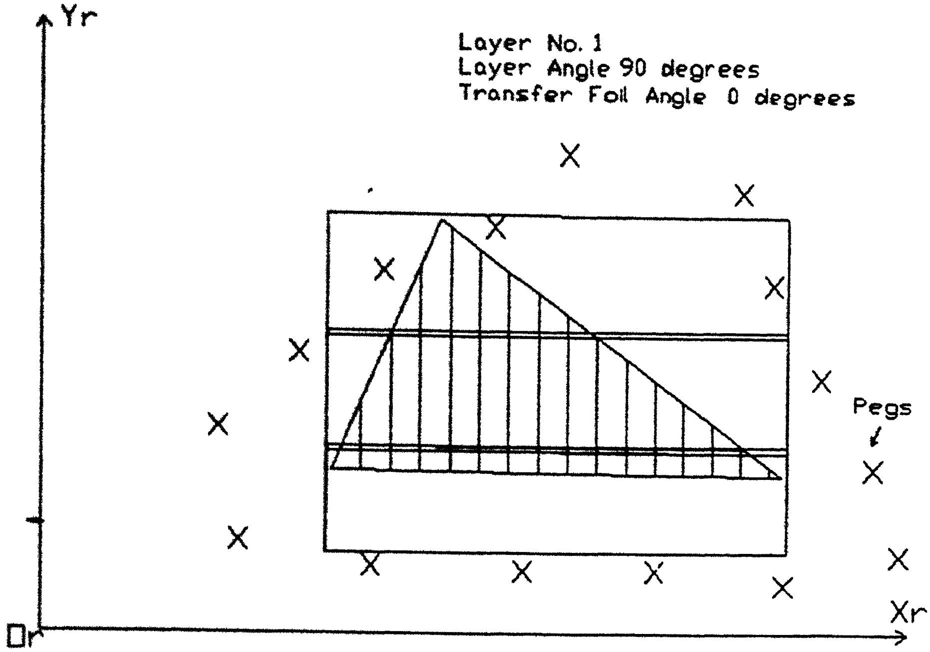

Figure 4.14 : Transfer foil layout for layer no. 1

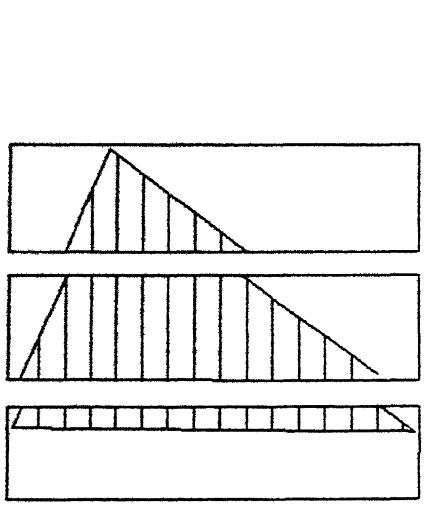

Figure 4.15 : Individual transfer foils for layer no. 1

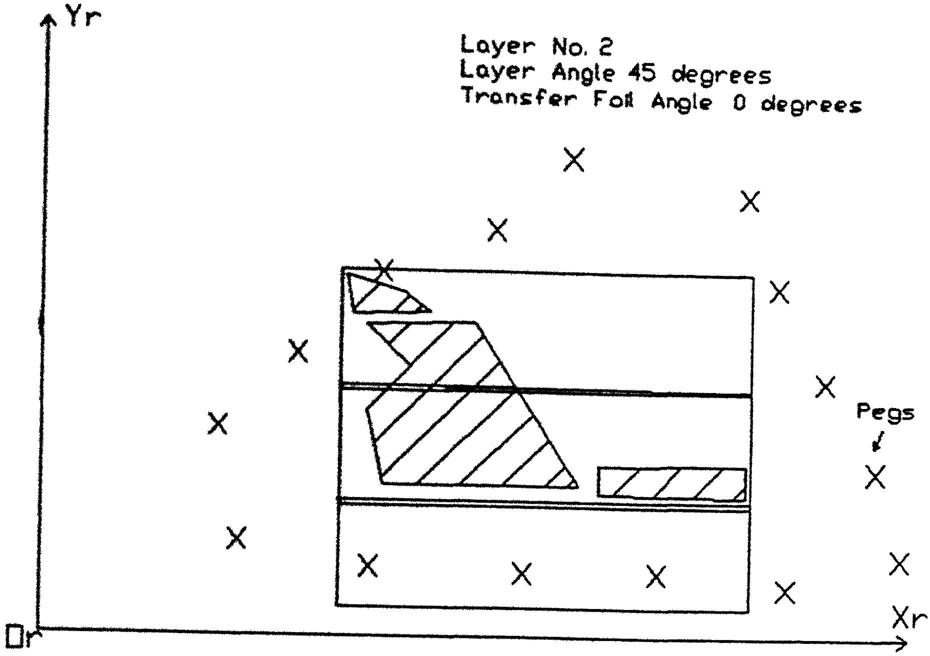

Figure 4.16 Transfer foil layout for layer no 2

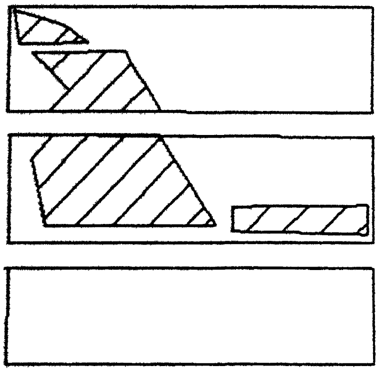

Figure 4.17 Individual transfer foils for layer no. 2

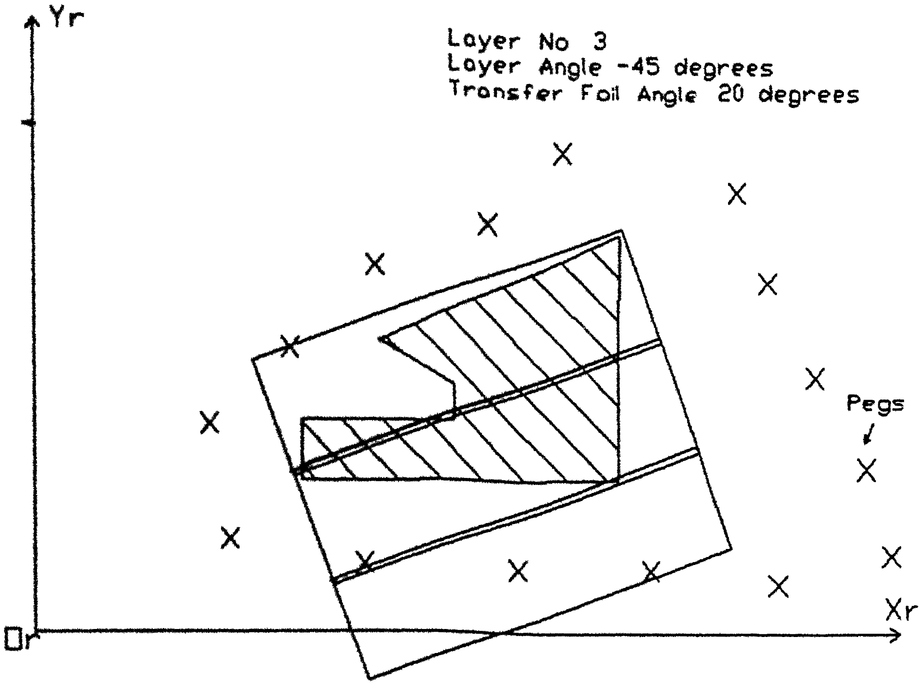

Figure 4.18 : Transfer foil layout for layer no 3

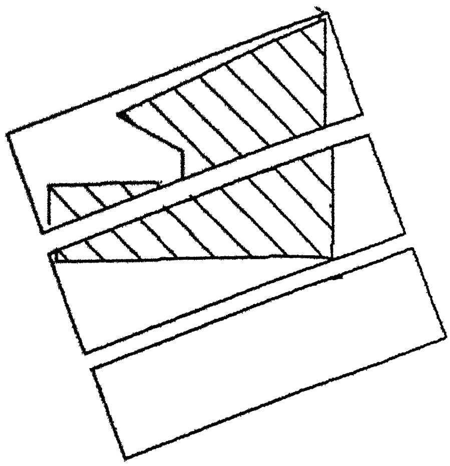

Figure 4.19 Individual transfer foils for layer no. 3

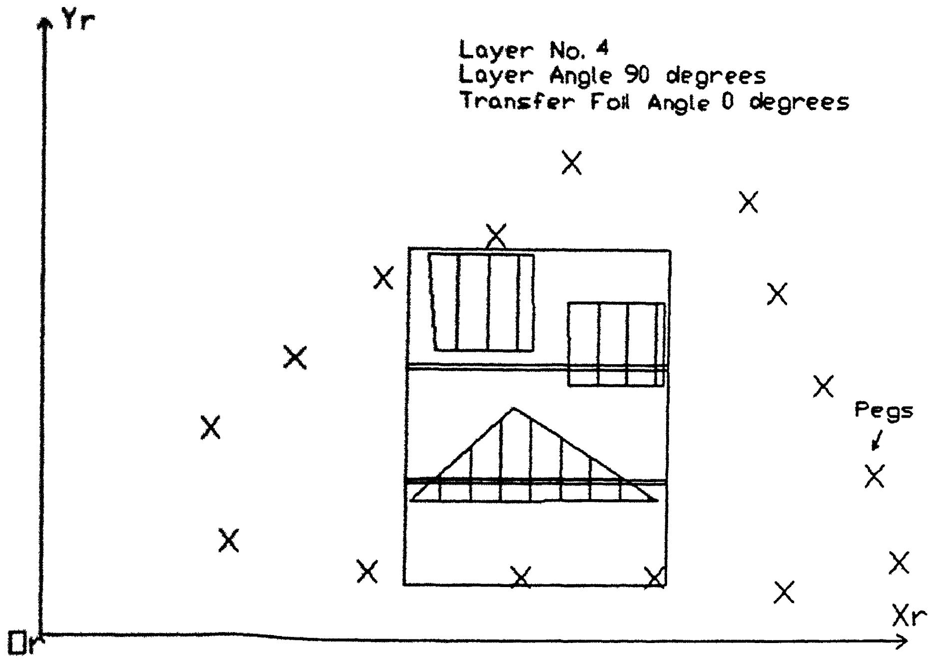

Figure 4.20 Transfer foil layout for layer no 4

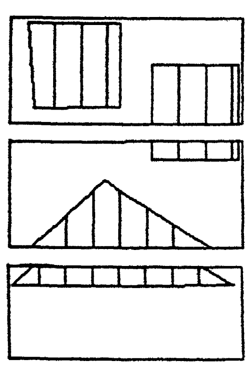

Figure 4.21 Individual transfer foils for layer no. 4

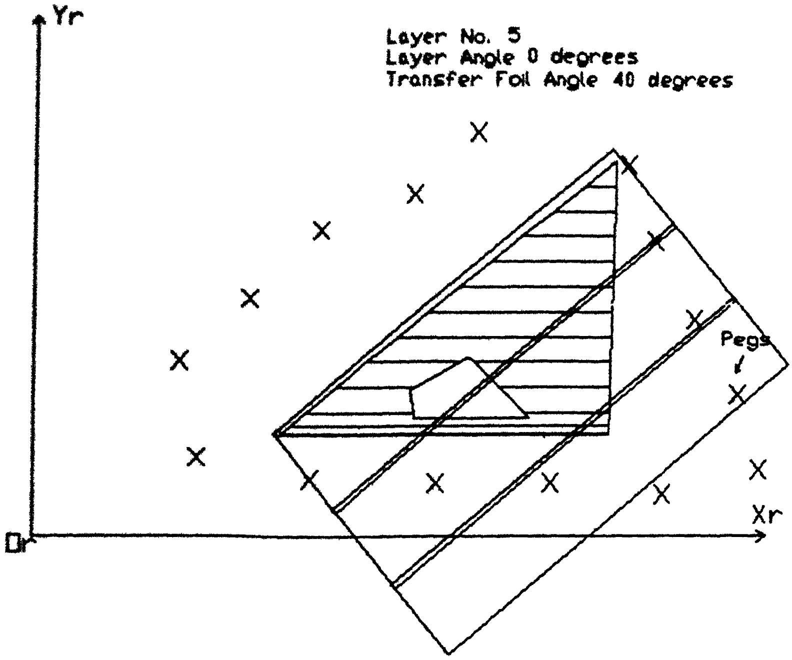

Figure 4.22 Transfer foil layout for layer no 5

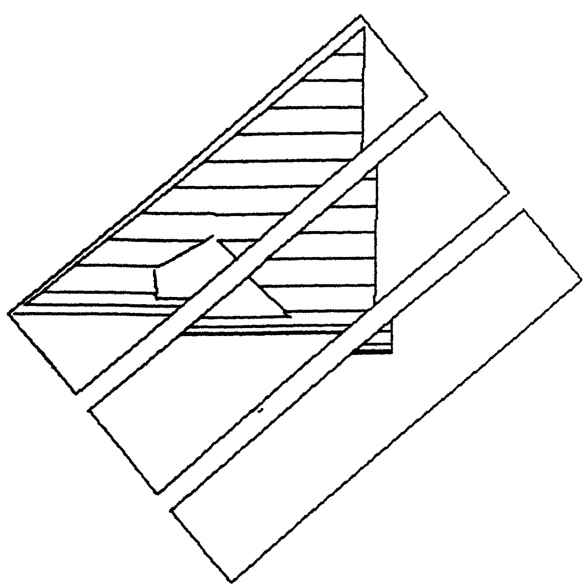

Figure 4.23 Individual transfer foils for layer no. 5

Chapter 5

CONCLUSION

5.1 Technical Summary

Design of composite components, like an aircraft wing, consists of several layers of composite material stacked in different orientations. Pre-impregnated composite materials are available in the form of tapes of standard widths. There is a lot of wastage in the process of cutting various layers from available tapes. In addition to this the accuracy of cutting and nesting is also poor in the conventional methods.

In the present work, an approach has been discussed for computer aided manufacturing of composite components. The whole process has been decomposed into many sub processes. After surface design, various layers are developed into flat patterns. The process of manufacturing of such components involves cutting of a number of pieces from unidirectional (UD) composite material tapes of standard widths. Cut pieces are nested on various tapes automatically and interactively. Cut pieces are then transferred to the mould using transfer foils. Mylar sheets are used as transfer foils.

For the complete process four different modules i.e. input of data for various layers of components, cut piece generation (strip layout), nesting of cut pieces, generation of data for transfer foil, have been identified. A computer software has been developed implementing all four above mentioned modules. Apart from this, there are input/output facilities and file formats. The layer

geometries for a composite component can be inputed to the software in a DXF file format or some other specified format. Tape widths spread on a given layer, to cover the entire layer, considering various constraints. Cut pieces thus formed are nested on the respective tapes so as to make maximum utilization of tapes while cutting cut pieces from them. Optimum transfer foil dimensions are found out for each layer using the pegs positions in the tool/mould. DXF file output is generated for cut pieces nested on tapes, which can be used for NC code generation for NC cutting of cut pieces from tapes.

The software package developed is menu driven and sufficient user interaction has been provided for flexible allocation of tapes for zones, arrangement of smaller cut pieces and modifying transfer foil assembly. A suitable database and a proper coding scheme has been considered in the program. The program has been tested for several examples.

5.2 Scope for Future Work

There is a lot of scope for future work in the software developed, under present work, to make the process of computer aided manufacturing of composite components fully automated.

Modules for flat pattern development of component surfaces [as discussed in Section 2.3] and for CNC code generation for cutting nested cut pieces from UD tape, are yet to be added to the software. Also a suitable interface has to be developed for plotting the cut piece geometries on mylar sheets.

In the nesting module of the present software cut pieces having the width equal to the available tape widths are neste

automatically (Section 3.2) but small pieces are allocated interactively on tapes. With some modifications in the present algorithm for nesting, these pieces can also be nested automatically, along with other pieces. This will make the nesting process faster.

REFERENCES

Composites

[1]. Jones, Robert M.,
"Mechanisms of Composites Materials",
McGraw-Hill, New York, 1976.

[2]. Agarwal, B.D. & Broutman, L.J.,
"Analysis and Performance of Fiber Composites",
John Wlley, 1980.

[3]. Special Report 799,
"Automating Composites Fabrication",
American Machinist and Automated Manufacturing, Nov 1987.

[4]. Donald, R.D.,
"The Challenge of Manufacturing Composites",
Machine Design, Oct 1987, pp. 92-98.

[5]. Hans Borstell,
"Manufacturing Processes: Aerospace",
Composites, ASM International,
Engineered Materials Hand-Book, Vol. 1, 1988, pp. 560-671.

Surface Design

[6]. Adams, R., 1989,
"Mathematical Elements of Computer Graphics",
2nd ed., McGraw-Hill, New York.

[7]. French, Thomas E., Vierck, Charles J. and Foster Rober J.,
"Graphic Science and Design",
Fourth Ed., McGraw-Hill, New York, 1984.

[8] Gurunathan, B.,
"Computer Aided Methods for Development of Certain Class of Ruled Surfaces",
Ph.D. Thesis submitted to the Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, February 1986.

[9]. Ramulu, P.B.,
"Computer-aided Methods for Development of Surfaces",
M-Tech Thesis submitted to the Department of Mechanical Engineering, Indian Institute of Technology Kanpur, June 1982.

[10]. Dhande, S.G., Gurunathan, B.,
"Algorithms for Development of Certain Class of Ruled Surfaces",
Computers and Graphics, Vol. 11, No. 2, 1987, pp. 105-112.

Nesting

[11]. Adamwicz, M. and Albano, A.,
"Nesting two-dimensional shapes in rectangular modules",
Computer Aided Design, Vol. 8, No. 1, Jan 1976, pp. 27-33.

[12]. Segenreich, S.A.,
"Optimal nesting of general plane figures: A Monte Carl
heuristical approach",
Computers and Graphics, Vol.10, No. 3, 1986, pp. 229-237

[13]. William W. Chow,
"Nesting of Single Shape on a Strip",
International Journal of Production Research, Vol. 1
No. 4, 1979, pp. 305-322.

[14]. Amaral, C., Bernardo, J., & Jorge, J.,

"Marker-Making using Automatic Placement of Irregular Shapes for the Garment Industry",
Computers and Graphics, Vol. 14, No. 1, 1990, pp. 41-46.

Transfer Foil

[15]. Technical Report,
Shama Rao, N.,
"Transfer Foil Concepts in Manufacturing of Composite Components for Aircraft Structures",
submitted to HAL Bangalore, 1989.

Others

[16]. Amitabh, G. & Mallik, A.K.,
"Manufacturing Science",
E.W.Publishers, 1978.

[17]. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.,
Network Flows - Theory, Algorithms and Applications,
Prentice-Hall, Inc. Englewood, Cliffs, NJ, USA, to be published in 1992-1993.

APPENDIX I

```
                    /*** type declarations ***/


/* vertex */

typedef struct {
   float  x,y,z;
}  vertex;


/* vertex list (polygon) */

typedef struct vrtx_lis {
   vertex          v;                      /* this vertex */
   struct          vrtx_lis  *nxtvert;     /* next vertex */
}  vertex_list;



/* data base structure for cut piece */

typedef struct cpdat {
  int              component;          /* component no */
  int              layerno;            /* layer number of
                                          cut piece */
  int              zoneno;             /* zone no */
  int              cpno;               /* cut-piece no w.r.t
                                          zone */
  float            cpwidth;            /* cut-piece width */
  int              row;                /* cut-piece row no */
  int              col;                /* cut-piece column no */
}  cp_data;




/* polygon list */

typedef struct poly_lis {
   vertex_list           *vlis;       /* list of vertices of
                                         this polygon */
   int                   polyno;      /* polygon number */
   struct   poly_lis     *nxtpoly;    /* next polygon */
   struct   poly_lis     *holes;      /* list of holes in
                                         the poly */
   cp_data               *cpdata;     /* cut piece data for
                                         nxf output */
}  polygon_list;

/* allocated polygon */
```

```
typedef struct alc_poly {
   polygon_list        *poly;          /* polygon */
   int                 polyno;         /* polygon number */
   float               xmin, ymin, xmax, ymax;
                                       /* poly extents in WC */
   int                 layout_size;       /* layout array size */
   double              *poly_strips       /* strip layout for the
                                             poly */
   polygon_list        *divplis;          /* list of polys divided
                                             across strips */
   struct  alc_poly    *nxtpoly;          /* next polygon */
}  aloced_plis;
```

```
/* list of layers of polygons */

typedef struct lyr_lis {
   int              layerno;              /* layer number */
   int              bundleno;             /* bundle no of layer */
   float          xmin, ymin, xmax, ymax;  /* layer window in WC */

   polygon_list   *rawplis;                  /* raw polygon list */
   aloced_plis    *alocpoly;                   /* strip-allocated
                                                  poly list */
   polygon_list   *procplis;                   /* processed polygon
                                                  list */
   float          angle_d;                     /* angle of layer in
                                                  degrees */
   int            layout_size;                 /* total layout
                                                  array size */
   double         *strip_layout;               /* total strip layout
                                           in this layer (unused) */
   struct lyr_lis  *nextlayer;                /* ptr to next layer */

   struct lyr_lis  *prevlayer;                   /* ptr to previous
                                                   layer */
}  layer_lis;
```

Global Variables

Several global variables are maintained in the software. They are meant for storing data belonging to current layer being processed and some settings.

```
                    /*** global declarations ***/

#define TRUE  1
#define FALSE 0

polygon_list     *Poly_header;              /* header of polygon list
*/
polygon_list     *Last_poly;               /* last polygon entered */
aloced_plis      *Alocatd_plis;             /* list of ud-tape
                                               allocations for each
                                               poly in the Poly_header
                                            */
aloced_plis      *Last_aloc_poly;           /* last polygon in the
                                               allocated plis */

polygon_list     *Divided_plis;            /* list of polygons divided
                                               across strips in one
                                               layer */

/* layers */

layer_list       *Layer_header;             /* header of layer list */

layer_list       *Last_layer;               /* last layer of list */
layer_list       *Current_layer;             /* current layer active
                                               (Must not be header)*/
int              Component_no;
                                             /* no of component
                                                being built */
int              No_of_layers;
                                             /* no of layers
                                                defined so far */
int              Next_layer_no;              /* indicates next
                                                default layer no */
int              Layers_for_stagger;
                                             /* no of layers for
                                                staggerconsiderations

*/
float            Stagger_allowance;
                                             /*   stagger   allowance

between
                                                layers in mm */

char             Input_file_dir[40];
                                              /* directory name
                                                 containing input files
                                              */

float            Layer_angle;
                                              /* layer orientation w.r.t

                                                 global frame */

#define          STRIP_TYPES     4          /* total no of strip types
```

```
                                               25, 75, 150, 300 units
                                            */

double          *Forbidden_bndry;
                                            /* array of boundaries
                                               from which next layer
                                               boundaries are
                                               forbidden */

double          strip_available[STRIP_TYPES]
                                           /* array of strip sizes */

double          cur_strip_layout[MAX_STRIP_BOUNDS];
                                           /* size of each strip at
                                              the boundaries in
                                              current layer */

Auto_Alloc_Width_Num                     /* current tape width number
                                            on which automatic
                                            nesting is done */

Interact_Width_Num                        /* width number of the cut
                              pieces which are nested interactively */
```

Vertices of a cut piece are stored in a link list of 'Rectype'

```
typedef struct record {
        float  x, y, z;                   /* vertex */
        struct record *next;             /* pointer to next vertex */
} Rectype;


typedef struct {
        char    cpcode[20];               /* code of the cut piece */
        float   width;                   /* width of the cut piece */
        Rectype * start;                  /* link list of vertices */

        int     ALLOC_DONE;              /* ALLOC_DONE is set TRUE if
                                          the cut piece is allocated */
} data_type;

typedef struct {
        float     width;                  /* width of UD-tape */
        int       no_of_pieces;           /* number of pieces of a
                                             particular width */
        int       no_of_pieces_on_tape;/* number of allocated
                                             pieces on the tape */
        float     tape_length_consumed;
        float     total_polys_area;          /* area of all the
                                                allocated cut pieces */
        data_type  * str;             /* pointer to array containing
                                         cut pieces data corresponding
                                         to a particular tape */
```

APPENDIX II

Input Data for Illustrative Example (Section 4.4)

```
layer no. 1    layer orientation 90
no. of zones 1
zone no. 1
vertices 3
    x                  y
6376.116699    4103.853027
9902.159180    1564.635132
5275.502441    1615.623413


layer no. 2    layer orientation 45.0
no. of zones 3
zone no.  1
vertices 6
    x              y
5815.618652    1472.856201
5652.564453    2217.285645
6090.771973    2665.982666
5652.564453    3084.086914
6732.797363    3104.482178
7772.266602    1472.856201
zone no.  2
vertices 4
    x                  y
5428.365234    3573.574707
6029.626953    3400.214355
6294.589844    3206.458740
5499.701660    3186.063477
zone no.  3
vertices 4
    x                  y
7965.893066    1676.8094480
9494.524414    1697.204712
9484.333008    1370.879517
7965.893066    1401.472534


layer no. 3    layer orientation -45.0
no. of zones 1
zone no.  1
vertices 7
    x                  y
8251.237305    3940.690430
8241.046875    1513.646851
5051.303223    1483.053833
5051.303223    2074.518311
6590.125000    2084.715820
6590.125000    2441.634033
5836.000488    2859.738281
```

layer no. 4 layer orientation 90.0
no. of zones 3
zone no. 1
vertices 3

x	y
6284.398926	1564.635132
7313.677246	2502.820068
8770.972656	1574.832764

zone no. 2
vertices 4

x	y
7853.793457	2737.366211
7843.602539	3573.574707
8811.735352	3573.574707
8811.735352	2737.366211

zone no. 3
vertices 4

x	y
6518.789062	3073.889160
6437.261719	4042.666992
7476.730957	4042.666992
7486.921875	3084.086914

layer no. 5 layer orientation 0.0
no. of zones 2
zone no. 1
vertices 3

x	y
5316.265625	1177.124023
9331.470703	4430.178223
9270.325195	1177.124023

zone no. 2
vertices 4

x	y
6946.805664	1370.879517
6906.041992	1697.204712
7599.021484	2084.715820
8322.574219	1370.879517